AMENDMENTS TO THE SPECIFICATION

Please replace the Title with "MICROPROCESSOR WITH SELECTIVELY OVERRIDING FIRST AND SECOND PREDICTIVE CALL/RETURN STACKS".

Please delete the section entitled "SUMMARY" in its entirety are substitute the following section therefor:

SUMMARY OF THE INVENTION

The present invention provides a branch prediction method and apparatus that makes efficient use of chip real estate, but also provides accurate branching early in the pipeline to reduce branch penalty. Accordingly, in attainment of the aforementioned object, it is a feature of the present invention to provide an apparatus in a processor for speculatively performing a return instruction. The apparatus includes first and second physically distinct call/return stacks, for providing first and second return addresses, respectively. Each of the first and second call/return stacks is configured to store a plurality of return addresses. The apparatus also includes a comparator, coupled to the first and second call/return stacks, for comparing the first and second return addresses prior to the return instruction reaching an execution stage of a pipeline of the processor. The execution stage is configured to finally resolve the return instruction. The apparatus also includes control logic, coupled to the comparator, for controlling the processor to branch to the first return address. The control logic subsequently controls the processor to branch to the second return address if the comparator indicates the first and second return addresses do not match.

In another aspect, it is a feature of the present invention to provide a microprocessor for predicting return instruction target addresses. The microprocessor includes an instruction cache, for generating a line of instruction bytes selected by a fetch address, the fetch address received from an address bus. The microprocessor also includes address selection logic, coupled to the address bus, for selecting the fetch address and providing the fetch address on the address bus. The microprocessor also includes a branch target address cache (BTAC), coupled to the address bus, for caching indications of previously executed return instructions and for providing one of the indications in response to the fetch address. The microprocessor also includes a first call/return stack, coupled to the BTAC, for providing a first return address to the address selection logic in response to the one of the indications. The first call/return stack is configured to store a plurality of return addresses. The microprocessor also includes decode logic, coupled to the instruction cache, for decoding the line of instruction bytes. The microprocessor also includes a second call/return stack, coupled to the decode logic, for providing a second return address to the address selection logic in response to the decode logic indicating that a return instruction is present in the line of instruction bytes. The second call/return stack is configured to store a plurality of return addresses. The second call/return stack is physically distinct from the first call/return stack. The microprocessor also includes an execution stage, coupled to the decode logic, for finally resolving return instructions. The first and second call/return stacks provide the first and second return addresses to the address selection logic prior to the return instruction reaching the execution stage.

In another aspect, it is a feature of the present invention to provide a method for speculatively branching a microprocessor to a target address of a return instruction. The microprocessor includes an execution stage for finally resolving the return instruction. The method includes generating a first target address by a first call/return stack. The first call/return stack is configured to store a plurality of return addresses. The method also includes branching to the first target address and generating a second target address by a second call/return stack subsequent to the branching to the first target address. The second call/return stack is also configured to store a plurality of return addresses. The second call/return stack is physically distinct from the first call/return stack. The method also includes comparing the first and second target addresses prior to the return instruction reaching the execution stage. The method also includes branching to the second target address if the first and second target addresses do not match.

In another aspect, it is a feature of the present invention to provide a microprocessor for predicting return instruction target addresses. The microprocessor includes an instruction cache, for providing a line of instructions in response to a fetch address received on an address bus. The microprocessor also includes a multiplexer, having a plurality of inputs, configured to select one of the plurality of inputs for provision on the address bus as the fetch address to the instruction cache. The microprocessor also includes a speculative branch target address cache (BTAC), coupled to the address bus, for indicating a speculative presence of a return instruction in the line of instructions. The microprocessor also includes a speculative call/return stack, coupled to the speculative BTAC, for providing a speculative return address to a first of the plurality of multiplexer inputs in response to the speculative BTAC indicating the speculative presence of the return instruction. The speculative call/return stack is configured to store a plurality of return addresses. The microprocessor also includes decode logic, configured to receive and decode the line of instructions. The microprocessor also includes a non-speculative call/return stack, coupled to the decode logic, for providing a non-speculative return address to a second of the plurality of multiplexer inputs in response to the decode logic indicating that the return instruction is actually present in the line of instructions. The speculative call/return stack is configured to store a plurality of return addresses. The non-speculative call/return stack is physically distinct from the speculative call/return stack. The microprocessor also includes a comparator, coupled to the speculative and non-speculative call/return stacks, for comparing the speculative and non-speculative return addresses prior to the return instruction reaching an execution stage of a pipeline of the processor. The execution stage is configured to finally resolve the return instruction. The multiplexer selects the speculative return address in a first instance, and selects the non-speculative return address in a second instance subsequent to the first instance if the comparator indicates that the speculative and non-speculative return addresses do not match.

An advantage of the present invention is that it potentially reduces the branch penalty by enabling the processor to branch on a return instruction without having to wait until the return instruction is decoded in contrast to conventional approaches having only a single call/return stack that branch later in the pipeline. Furthermore, the present invention potentially provides more accurate return instruction prediction than a speculative branch target address cache providing the return address since the call/return stack accounts for the possibility of multiple return paths.